UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/939,232	08/24/2001	William Joseph Armstrong	IBM / 182	4082

26517 7590 03/07/2008

WOOD, HERRON & EVANS, L.L.P. (IBM)
2700 CAREW TOWER
441 VINE STREET
CINCINNATI, OH 45202

EXAMINER	
PROCTOR, JASON SCOTT	

ART UNIT	PAPER NUMBER
2123	

MAIL DATE	DELIVERY MODE
03/07/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte WILLIAM JOSEPH ARMSTRONG, CHRIS FRANCOIS,
and NARESH NAYAR

Appeal 2007-3352
Application 09/939,232
Technology Center 2100

Decided: March 7, 2008

Before LANCE LEONARD BARRY, HOWARD B. BLANKENSHIP, and CAROLYN D. THOMAS, *Administrative Patent Judges*.

THOMAS, C., *Administrative Patent Judge*.

DECISION ON APPEAL

I. STATEMENT OF THE CASE

Appellants appeal under 35 U.S.C. § 134 from a Final Rejection of claims 1-19 and 21 entered February 15, 2006. We have jurisdiction under 35 U.S.C. § 6(b).

We reverse.

A. INVENTION

Appellants invented a method, an apparatus, and a program product for coordinating the distribution of central processing units (CPUs) among logically-partitioned virtual processors. (Spec., Abstract.)

B. ILLUSTRATIVE CLAIM

The appeal contains claims 1-19, and 21. Claims 1, 11, and 19 are independent claims. As best representative of the disclosed and claimed invention, claim 1 is reproduced below:

> 1. A method for yielding a virtual processor within a logically partitioned data processing system, wherein the system supports a plurality of partitions, a first of which includes a plurality of virtual processors used to schedule threads and that share at least one CPU, and wherein the system further includes a hypervisor configured to assign and dispatch the CPU to the plurality of virtual processors, the method comprising:
>
> requesting with a yielding virtual processor a yield of the CPU upon which the virtual processor is executing, including designating a target virtual processor from among the plurality of virtual processors; and
>
> switching-in the target virtual processor for execution by the CPU in response to the requested yield.

C. REFERENCES

The references relied upon by the Examiner in rejecting the claims on appeal are as follows:

Greene	US 5,404,563	Apr. 4, 1995
Bitar	US 5,872,963	Feb. 16, 1999

Abraham Silberschatz and Peter Baer Galvin, *Operating System Concepts,* John Wiley & Sons, Inc., 5th Edition, 74-75 (1999) (hereinafter "Silberschatz").

D. REJECTION

The Examiner entered a Final Rejection on February 15, 2006 with the following rejection, which is before us for review:

Claims 1-19, and 21 are rejected under 35 U.S.C. § 103(a) as obvious over Greene, Bitar and Silberschatz.

II. PROSECUTION HISTORY

Appellants appealed from the Final Rejection and filed an Appeal Brief (Br.) on July 19, 2006. The Examiner mailed an Examiner's Answer (Ans.) on October 31, 2006. No Reply Brief was filed.

III. ISSUE

Whether Appellants have shown that the Examiner erred in rejecting claims 1-19 and 21 as obvious over the combination of Greene, Bitar and Silberschatz.

IV. FINDINGS OF FACT

The following findings of fact (FF) are supported by a preponderance of the evidence.

Bitar

1. Bitar discloses that "[t]he architecture is typically implemented by building a user-level scheduler that manages the switching of the user-level threads onto the kernel-level threads. A kernel scheduler is then responsible for scheduling the virtual processor onto physical processors." (Col. 4, ll. 34-39.)

2. Bitar discloses that "[a] virtual processor may be a process, . . . a kernel thread, . . . or some other abstraction." (Col. 1, ll. 34-36.)

3. Bitar discloses that "[t]he operating system will provide a unit of scheduling, a virtual processor, to which a user-level thread will be mapped. . . .This virtual processor will in turn be mapped to a physical processor by the operating system scheduler. Conceptually, it is useful to distinguish the user-level thread from the virtual processor." (Col. 1, ll. 27-32.)

4. Bitar discloses that "[w]hile the kernel maintains its traditional responsibility of scheduling virtual processors onto physical processors, the

threads library now has to schedule user-level threads onto virtual processors." (Col. 4, ll. 56-59.)

V. PRINCIPLES OF LAW

Appellants have the burden on appeal to the Board to demonstrate error in the Examiner's position. *See In re Kahn*, 441 F.3d 977, 985-86 (Fed. Cir. 2006) ("On appeal to the Board, an applicant can overcome a rejection [under § 103] by showing insufficient evidence of prima facie obviousness or by rebutting the prima facie case with evidence of secondary indicia of nonobviousness.") (quoting *In re Rouffet*, 149 F.3d 1350, 1355 (Fed. Cir. 1998)).

VI. ANALYSIS

Common Feature In All Claims

Our representative claim, claim 1, recites, *inter alia, "requesting with a yielding virtual processor a yield of the CPU . . ., including designating a target virtual processor. . .; and switching-in the target virtual processor for execution by the CPU."* Independent claims 11 and 19 recite similar limitations. Thus, the scope of each of the independent claims includes a request for a yield of the CPU, wherein the request designates a target virtual processor and reassigning control of the CPU from the yielding virtual processor to the target virtual processor.

The Board's Claim Construction

"Our analysis begins with construing the claim limitations at issue." *Ex Parte Filatov*, No. 2006-1160, 2007 WL 1317144, at *2 (BPAI 2007). Claims are given their broadest reasonable construction "in light of the specification as it would be interpreted by one of ordinary skill in the art." *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004).

The Examiner found that "Appellants' specification defines a virtual processor as a logical thread of execution." (Ans. 9). The Examiner further found that "[t]he Bitar reference defines a virtual processor as equivalent to a kernel thread." *Id.* As a result, the Examiner concluded that "a virtual processor is a thread." *Id.* We disagree with this conclusion.

Initially, we find that the Specification actually discloses that "virtual processors *act* as logical threads of execution for a host partition. As such, the virtual processors can separately execute instructions, while sharing resources." (Emphasis added) (Spec., 2, ll. 15-17.)

In other words, while the Specification identifies a similarity between a "virtual processor" and "threads of execution", we find that the Specification does not expressly define a virtual processor as a logical thread of execution but instead identifies similar functions that can be performed by both "threads" and "virtual processors", i.e., separately executing instructions while sharing resources. Thus, contrary to the Examiner's findings, we find that the Appellants have not limited a virtual processor to merely being a thread of execution.

In addition, we find that Bitar shows varying definition for threads by identifying at least two types of threads, i.e., user-level threads and kernel-level threads (FF 1). Thus, while Bitar equates a "virtual processor" with a "kernel-level thread" (FF 2), Bitar also distinguishes a "virtual processor" from "user-level threads" (FF 3). Bitar further discloses that user-level threads are scheduled onto virtual processors (FF 4), implying that virtual processor are used to execute/run threads, specifically user-level threads.

As such, as disclosed by Bitar, while a virtual processor can be seen as a kernel-level thread, a virtual processor cannot reasonably be identified as a "thread" in and of itself without some type of qualifying identifier because threads are of various types. Greene and Silberschatz provide no additional clarification between threads and virtual processors.

Therefore, we disagree with the Examiner's broad conclusion that a virtual processor is a generic thread.

35 U.S.C. § 103(a): Claims 1-19 and 21

"Having determined what subject matter is being claimed, the next inquiry is whether the subject matter would have been obvious." *Ex Parte Massingill*, No. 2003-0506, 2004 WL 1646421, at *3 (BPAI 2004). The question of obviousness is "based on underlying factual determinations including . . . what th[e] prior art teaches explicitly and inherently . . . ." *In re Zurko*, 258 F.3d 1379, 1383 (Fed. Cir. 2001) (citing *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966); *In re Dembiczak*, 175 F.3d 994, 998

(Fed. Cir. 1999); *In re Napier*, 55 F.3d 610, 613 (Fed. Cir. 1995)). "In rejecting claims under 35 U.S.C. § 103, the examiner bears the initial burden of presenting a *prima facie* case of obviousness." *In re Rijckaert*, 9 F.3d 1531, 1532 (Fed. Cir. 1993) (citing *In re Oetiker*, 977 F.2d 1443, 1445 (Fed. Cir. 1992)). "'A *prima facie* case of obviousness is established when the teachings from the prior art itself would appear to have suggested the claimed subject matter to a person of ordinary skill in the art.'" *In re Bell*, 991 F.2d 781, 783 (Fed. Cir. 1993) (quoting *In re Rinehart*, 531 F.2d 1048, 1051 (CCPA 1976)).

The Examiner found that there is a distinction between user-level threads and a kernel thread (Ans. 9). The Examiner continues by finding that Bitar teaches that threads which have finished their work can transfer control of their processors to the preempted threads, thus resuming the preempted thread (Ans. 4).

Appellants contend that "[t]he objective of *Bitar* is to achieve switching between user threads without involving the scheduling of virtual processors (col. 5, lines 31-33 and lines 55-58). *Bitar* teaches away from using a virtual processor, or kernel, and associated scheduling during thread switching for efficiency reasons (col. 5, lines 14-18, 31-38, 55-58 and col. 12, line 24)." (Suppl. Br. 8-9.) Appellants further contend that the "Examiner's attempt to expand *Bitar's* definition of execution entities to include virtual processors is improper and contrary to the plain text of *Bitar*." (Suppl. Br. 9.) We agree.

Bitar discloses at column 11, lines 1-10 the following:

> In a nanothreaded model formed according to the present invention, instead of spinning, the threads that have completed their work can query the preempted bit vector to determine that the other threads working on the loop have been preempted; if so, the threads which have finished work can transfer control of their respective processors to the preempted threads, thus resuming them. In one embodiment, the transfer of processor is accomplished in the nanothreads model by a resume interface and requires no kernel intervention.
> (Col. 11, ll. 1-10.)

In other words, Bitar contemplates transferring a processor from one thread to another thread without the assistance of the kernel. Thus, while Bitar discloses a thread switching process using a resume interface, given our distinction *supra* regarding threads and virtual processors, we find that the Examiner's reliance on Bitar's thread-switching teachings has failed to establish that Bitar's thread-switching process is equivalent to a virtual processor requesting yield of a CPU, including designating a target virtual processor and switching-in the target virtual processor for execution by the CPU, as set forth in the present invention. The Examiner has also failed to establish that Greene and Silberschatz disclose the above noted features.

Therefore, we will *not* sustain and will instead reverse the Examiner's rejection under 35 U.S.C. § 103 for the same reasons as set forth above.

VII. CONCLUSIONS

We conclude that Appellants have shown that the Examiner erred in rejecting claims 1-19 and 21.

VIII. DECISION

In view of the foregoing discussion, we reverse the Examiner's rejection of claims 1-19 and 21.

REVERSED

rwk

WOOD, HERRON & EVANS, L.L.P. (IBM)
2700 CAREW TOWER
441 VINE STREET
CINCINNATI OH 45202